K-Nearest Neighbors Classifier

A K-Nearest Neighbors (KNN) classifier is a classification model that uses the nearest neighbors algorithm to classify a given data point.

The basic concept of K-nearest neighbor classification is to find a predefined number, i.e., the 'k' − of training samples closest in distance to a new sample, which has to be classified.

New samples will get their label from the neighbors itself. The KNN classifiers have a fixed user defined constant for the number of neighbors which have to be determined.

For the distance, standard Euclidean distance is the most common choice. The KNN Classifier works directly on the learned samples rather than creating the rules for learning.

from sklearn.datasets import \*

import pandas as pd

%matplotlib inline

from sklearn.neighbors import KNeighborsClassifier

import matplotlib.pyplot as plt

import numpy as np

#Load data of digit images

#From MNIST database, with 1797 images,

#digit\_d = pd.DataFrame(digit['data'][0:1600])

digit = load\_digits()

#Train with the first 1600 images

number\_img\_to\_train = digit['data'][:1600]

#Has the number as 0,1,2,...

number\_value = digit['target'][:1600]

#To show the image of the number

def Image\_display(i):

    plt.imshow(digit['images'][i],cmap = 'Greys\_r')

    plt.show()

#Muestra 0 y su imagen

#print(number\_value[0])

#Image\_display(0)

#Muestra 9 y su imagen

#print(number\_value[9])

#Image\_display(9)

#Train the moment to refer each image to a specific number

#=========================================================

#Create model

KNN = KNeighborsClassifier(20)

#Train model

KNN.fit(number\_img\_to\_train,number\_value)

#Create the K nearest neighbor classifier constructor

KNeighborsClassifier(algorithm ='auto', leaf\_size =30, metric ='minkowski',

   metric\_params= None, n\_jobs =1, n\_neighbors =20, p =2, weights= 'uniform')

# With the model trained, find the value for image 1725

# which is for the number 6

# Look for the nearest possible reference

test = np.array(digit['data'][1725])

#mostrara la imagen del numero 6

#Image\_display(1725)

test1 = test.reshape(1,-1)

#Busque el numero relacionado a la imagen MAS CERCANA

arr = KNN.predict(test1)

pos = arr[0]

print(number\_value[pos])

output: 6

|  |  |  |
| --- | --- | --- |
| Imagen 0 | Imagen 6 / 1725 | Imagen 9 |
| Image_display(0) | Image_display(6) | Image_display(9) |

Si digita

digit['target\_names']

Saldra

array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])